SOFTWARE CONTENT DOWNLOADING METHODS
IN RADIO COMMUNCATION NETWORKS

FIELD OF THE DISCLOSUREINVENTIONS

[0001]   The present disclosure relates inventions relate generally to communications between radio access networks and mobile terminals therein, and, more particularly, to methods for downloading software to multiple terminals in wireless communication networks, for example, in cellular communication networks.

BACKGROUND OF THE INVENTIONS

[0002]   The emergence of many new wireless technologies, including Wireless Application Protocol (WAP), Java 2 micro-edition programs (J2ME), mobile execution environment (MexE) software, Software Definable Radio (SDR), Terminal Management, among many others, require requires over-the-air downloading of terminal software to multiple terminals in wireless communication networks. It will soon be desirable to download software files as large as several Mbytes, and the trend is toward the transfer of even larger files.

[0003]   The over-the-air downloading of substantial software content in wireless communication networks, however, will likely impose significant burdens on the radio frequency spectrum resources allocated to network operators. In the future, as the code size of terminal software increases, network operators may be compelled to compensate users for software download air-time, for example with free air-time minutes, which is an undesirable cost overhead for the network operators.

[0004]   The various aspects, features and advantages of the present ~~inventions~~ disclosure will become more fully apparent to those having ordinary skill in the art upon careful consideration of the following Detailed Description ~~of the Invention~~ with the accompanying drawings, which are described below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005]   FIG. 1 is an exemplary schematic of a basic software download process flow diagram.

[0006]   FIG. 2 is an exemplary schematic of a wireless communication link block diagram corresponding to the basic software download process flow diagram of FIG. 1.

[0007]   FIG. 3 is a more particular exemplary schematic of a software download process flow diagram.

[0008]   FIG. 4 is an exemplary schematic of a wireless communication link block diagram corresponding to the more particular software download process flow diagram of FIG. 3.

[0009]   FIG. 5 is an exemplary wireless communication network in which the software download processes ~~of the present invention~~ may be practiced.

DETAILED DESCRIPTION ~~OF THE INVENTIONS~~

[00010]   The present disclosure provides ~~inventions provide a~~ spectrally efficient means for over-the-air (OTA) downloading of software objects, or

content, to multiple terminals in wireless communication networks, especially in networks where the population of terminals exceeds the number of unique software objects to be downloaded, for example, when all or several terminals in the network are to receive a software revision or upgrade. The invention is not limited to applications where only a single, common software object is transmitted to multiple users or terminals.

[00011]   The invention provides more generally for efficient simultaneous, or at least virtually simultaneous, downloading of multiple software objects to a plurality of terminals in wireless communication networks. Some terminals may receive some software content, and other terminals may receive other content. In some embodiments, the dynamics of the software content transmitted by the network and received by the terminals changes dynamically as downloads are completed and new terminals enter into the network, as discussed further below.

[00012]   Generally, the invention makes use of both shared (common) channels and dedicated (traffic) channels in hybrid, over-the-air software downloading schemes. The processes of the present inventions are implemented generally in several phases, otherwise referred to as communication exchanges between the terminals and network. For each phase, the communications between the network and terminals therein are allocated most efficiently to either a common channel or to a dedicated channel, depending upon the nature of the communication or exchange.

[00013]   The communication phases involving the exchange of terminal unique information are generally assigned to dedicated communication channels. Terminal unique information includes, for example, download initiation information exchanges, capability and non-repudiation information exchanges,

activation and billing information exchanges, and other communications that inherently involve the transmission of relative small quantities of data. These small data content exchanges are only exemplary. There could be other information exchanges not listed, and not all of the exemplary exchanges are required. These and other communications are well suited for point-to-point or dedicated channels, since the volume of data exchanged is relatively small and the spectral inefficiency of dedicated channels is insignificant, at least relative to that associated with the transmission of software content having file sizes on the order of 1 Mbyte or more.

[00014]   Other communications occurring on dedicated channels include those for which optimum error protection is required, for example the transmission of digital signatures and other error sensitive information. In some applications, the desire for providing error protection may outweigh the benefits provided by reduced bandwidth associated with transmission of broadcast information over common channels.

[00015]   The communication phases involving the transmission, or downloading, of substantial amounts of software objects or content from the network to one and preferably to multiple terminals in the network are assigned to common channels, which continuously stream downloadable content from the network to the terminals. In this way, the spectral requirement for the bandwidth intensive part of the download process is minimized. The data transfer typically involves relatively large quantities of data, e.g., terminal software content, sent primarily in the downlink direction to multiple terminals.

[00016]   An example of a common channel is the Packet Broadcast Control Channel (PBCCH) of a GPRS network. An example of a dedicated channel is the

4

Packet Data Traffic Channel (PDTCH) of a GPRS network. In the exemplary GPRS network implementation ~~the invention~~, one or more PBCCHs could be allocated for purposes of broadcast software downloading.

[00017]   The present <u>disclosure is</u> ~~inventions are~~ not limited to applications in GPRS networks, but may be applied instead to any wireless communication standard employing some type of over-the-air (OTA) software download. The initial commercial opportunities for application of the present inventions will likely be in higher tier and smart terminal product markets, but in the not too distant future the transfer of substantial amounts of software content to mobile terminals will be commonplace at many if not most wireless communication network service levels.

[00018]   In FIG. 1, the downloading of software from the network to a terminal is initiated at block 100 on a dedicated channel. The initiation may be prompted by the network or by the terminal. FIG. 2 generally characterizes the download initiation communications between the terminal 200 and the network 210 as a relatively small bi-directional data flow, which is allocated to a dedicated channel. The preliminary initiation exchanges, at least initially, will likely be transparent to the terminal user, although for some applications it could be user initiated and may require user input or response.

[00019]   In FIG. 1, at block 120, download capability exchange information is exchanged between the network and the terminal, and in FIG. 2 the information exchange is characterized generally as a relatively small bi-directional data flow, which as noted is also allocated to a dedicated channel.

[00020]   In FIG. 1, the data content download occurs at block 130. FIG. 2 illustrates this data transfer as a large asymmetrical data transfer from the network 210 to the terminal 200, preferably to multiple terminals, which are not illustrated.

[00021]   FIG. 1 illustrates, at block 140, a software installation step, which is a local process occurring at the terminal, as illustrated in FIG. 2. Generally, upon receipt of a software download, or after installation thereof by or at the terminal, the terminal transmits a software receipt confirmation notification to the network. This low data content confirmation communication preferably occurs on a dedicated channel.

[00022]   FIG. 1, at block 150, also illustrates the communication of non-repudiation information, and at At block 160, activation and billing information, which are both characterized by small bi-directional data flows, are preferably allocated to dedicated communication channels.

[00023]   The process flow diagram of FIG. 3 illustrates some of the same information communicated between the network and terminal as illustrated in FIG. 1. Particularly, in FIG. 3, the download initiation 300 and capability exchange 320, installation 340, and the non-repudiation exchange 350, and activation and billing exchange 360, which are not all essential, though typical in one form or another of a software content download transaction. Other exchanges, not illustrated, may also be included.

[00024]   FIG. 3 also illustrates, in the data transfer block 330, the communication or transmission of a digital signature 332 generally during the data transfer phase 330 from the network to the plurality of terminals on the dedicated communication channel. FIG. 4 illustrates, with spatial correspondence

relative to FIG. 3, whether the various exchanges in FIG. 3 are allocated to dedicated or common channels in the network.

[00025]   In the well-known Public Key Infrastructure (PKI), a digital signature is produced and transmitted by the sender, which is the network in most embodiments of the present inventions. More particularly, the file or data to be transferred is initially converted to a "message digest" with a hashing Hash function. The "message digest" is subsequently encrypted with the sender's private key, thus producing the digital signature. The PKI encryption application is only exemplary and the invention is not intended to be limited to any particular encryption scheme.

[00026]   In the present invention, generally, the digital signature may be transmitted to the terminals from the network on either dedicated channels or on a shared channel, since the digital signature is common information. Transmission over a dedicated channel will, by virtue of its bi-directional nature, provide far superior error protection for the digital signature compared to transmission over a common channel. Therefore, in a preferred application, the digital signature 332 is transmitted on a dedicated channel, as illustrated in FIGS. 3 and 4.

[00027]   The sender's public key can be installed in the terminal, for example, in phone software, in a number of ways. The public key may be programmed in the handset by the manufacturer, for example by installing a ROM chip. An alternative is for the network operator to program the public key into the terminal prior to selling the phone to a subscriber. Yet another alternative is to transmit the public key to the terminal, preferably via a dedicated channel. In this case, the public key could originate from a server controlled by either the equipment manufacturer, or from the network operator, or from a Certificate Authority (CA).

[00028]   In FIG. 3, upon receipt of the data transmitted during the data transfer 334, the public key, a copy of which resides on the terminal, enables the authentication or verification 336 of the digital signature 332 in a local process occurring at the terminal. The integrity of the data may also be verified.

[00029]   In some software download applications, the software content transmitted by the network comprises a plurality of different software files multiplexed on a shared communication channel for receipt by multiple terminals, thus providing different software content, which may be downloaded concurrently by different terminals.

[00030]   In applications where two or more software files are multiplexed on the shared communication channel, the software content may be dynamically adjusted on the shared communication channel, for example to more efficiently accommodate the requirements of the terminals receiving the software.

[00031]   In one application, the software content multiplexed on the shared communication channel is adjusted dynamically by adjusting a transmission time of each of the plurality of software files. Assume, for example, that the software download process is being managed for 1000 terminals in the network: 600 terminals require software object A; 300 terminals require software object B; and 100 terminals require software object C.

[00032]   In one embodiment, software object A will be transmitted on the shared channel 60 percent of the time, and software objects B and C will be transmitted 30 percent and 10 percent of the time, respectively. The exemplary temporal proportions can be adjusted dynamically to accommodate changes in the number of terminals requiring the different software objects, for example as active

8

download processes are completed and new downloads are initiated as indicated upon completion of activation and billing exchanges and new initiation exchanges discussed above.

[00033]   In another embodiment, the software content multiplexed on the shared communication channel is adjusted dynamically by adjusting the number of times each of the plurality of software files is transmitted.

[00034]   In another embodiment, the software content multiplexed on the shared communication channel is prioritized, for example, by giving higher priority to the transmission of content that generates greater amounts of revenue relative to the transmission of that which generates lesser amounts of revenue.

[00035]   In another embodiment, the software content multiplexed on the shared communication channel is prioritized by giving higher priority to the transmission of software content that is more essential over that which is less essential, for example, operating system updates may be prioritized over application or optional software updates. The software content multiplexed on the shared communication channel may also be adjusted dynamically based upon file size or content.

[00036]   In the exemplary network architecture 500 of FIG. 5, a radio device management server 510 manages the download process for all terminals 520 in the radio access network 530. In some implementations, the server 510 generates a continuous software download stream, which is mapped to the physical shared channel. The Radio Device Management Server may thus dynamically adjust the proportionality and/or priority of multiple software objects multiplexed onto the

shared channel. The software download may be managed alternatively by means other than a dedicated management server.

[00037]   The time multiplexed software download payloads can be fragmented using conventional packet transmissions. Packet protocols generally include headers having software checksums for positive identification of the encapsulated payload, fragment index counters for identifying the current fragment in the overall payload, a next transmission field that informs the terminal when the next fragment will be transmitted, as is known generally by those having ordinary skill in the art.

[00038]   The exemplary architecture of FIG. 5 includes a configuration management server 540 that contains a database that defines approved and disapproved hardware and software configurations. The configuration management server 540 is—may be managed for example by an equipment manufacturer, and made accessible by the Radio Device Management Server 510. The database on the configuration server 540 contains, for example, a unique identifier of the software ("type"), the software version indicator (software "revision"), and a cryptographic checksum ("checksum"), which collectively identify the software uniquely and allow verifying that it has been fetched correctly. This information can be presented to the Manufacturer's Software Download Server to fetch a copy of the designated software.

[00039]   In other embodiments, the exemplary architecture of FIG. 5 also includes a configuration management server 550 containing new software releases, including core software. Content from the server can be electronically signed by the manufacturer, thus allowing the terminal device to process the content according to security protocols running in the terminal device (e.g. MExE). The

configuration management server 550 is accessible by the radio device management server 510.

[00040]   While the present inventions and what is considered presently to be the best modes thereof have been described in a manner that establishes possession thereof by the inventors and that enables those of ordinary skill in the art to make and use the same inventions, it will be understood and appreciated that there are many equivalents to the exemplary embodiments disclosed herein and that myriad modifications and variations may be made thereto without departing from the scope and spirit of the inventions, which are to be limited not by the exemplary embodiments but by the appended claims.

[00041]   What is claimed is: